Assertion and Argumentation
Assignment 4

Clemens Beissel
4547330 Group 4

March 11, 2018

Exercise 1

a) Currently the heir to the throne is ...

b) Under the assumption that all daughters of Willem-Alexander are dead, then the heir would be ...

c) Remove the options that are not applicable...

- It was a lot of work but I managed to find a solution...
- I tried to find a solution, but I got stuck at ...
- I only looked at the case where the heir is a legitimate descendant of the King...
- ...

And this is the result:

```
Definition HeirToTheThrone (p:HON) :=
...
```

Exercise 2

a) The three statements are:

- On suitcase 1 is written "...".
- On suitcase 2 is written "...".
- On suitcase 3 is written "...".

b) According to the SAT-solver the gold is in suitcase ..., and the true statement is on suitcase ....

c) The following model has exactly two solutions.

(i) In natural language:

Anna says:
Bernie says:
Carl says:

(ii) The first model I used is:

```
[Obviously this is a wrong model for this exercise.]
(declare-const A Bool)
(declare-const B Bool)
(declare-const C Bool)
(declare-const D Bool)
(declare-const E Bool)
(declare-const F Bool)
(assert (= B (or E A)))
(assert (= C (and (not A) (not F))))
(assert (= D (and F A)))
(assert (= E (= F (not B))))
(assert (= F (or B C)))
(check-sat)
(get-model)
```

The result given by the SAT-solver is

```
[Obviously this is the solution to a wrong model for this exercise.]
sat
(model
  (define-fun A () Bool true)
  (define-fun B () Bool true)
  (define-fun F () Bool true)
  (define-fun E () Bool false)
  (define-fun D () Bool true)
  (define-fun C () Bool false)
)
```

The second model I used is:

...

The result given by the SAT-solver is

...

(iii) The model I used to prove that there are exactly two solutions is

...

The result given by the SAT-solver is

...

Exercise 3

a) I want to point out something strange in this part of the text:

$Definition computermouse :=$
$((cm\_power$
$->$
$cm\_sensor)$
$/\backslash$
$move\_mouse)$
$->$
$cursor\_move$
...

To be precise, I find it strange that in this case the mouse doesn't have to be connected to any computer port that has the required drivers installed to recognize the mouse. But that is a necessity for the mouse to work.

b) This is the device I added on page

`Wiring-copy1`

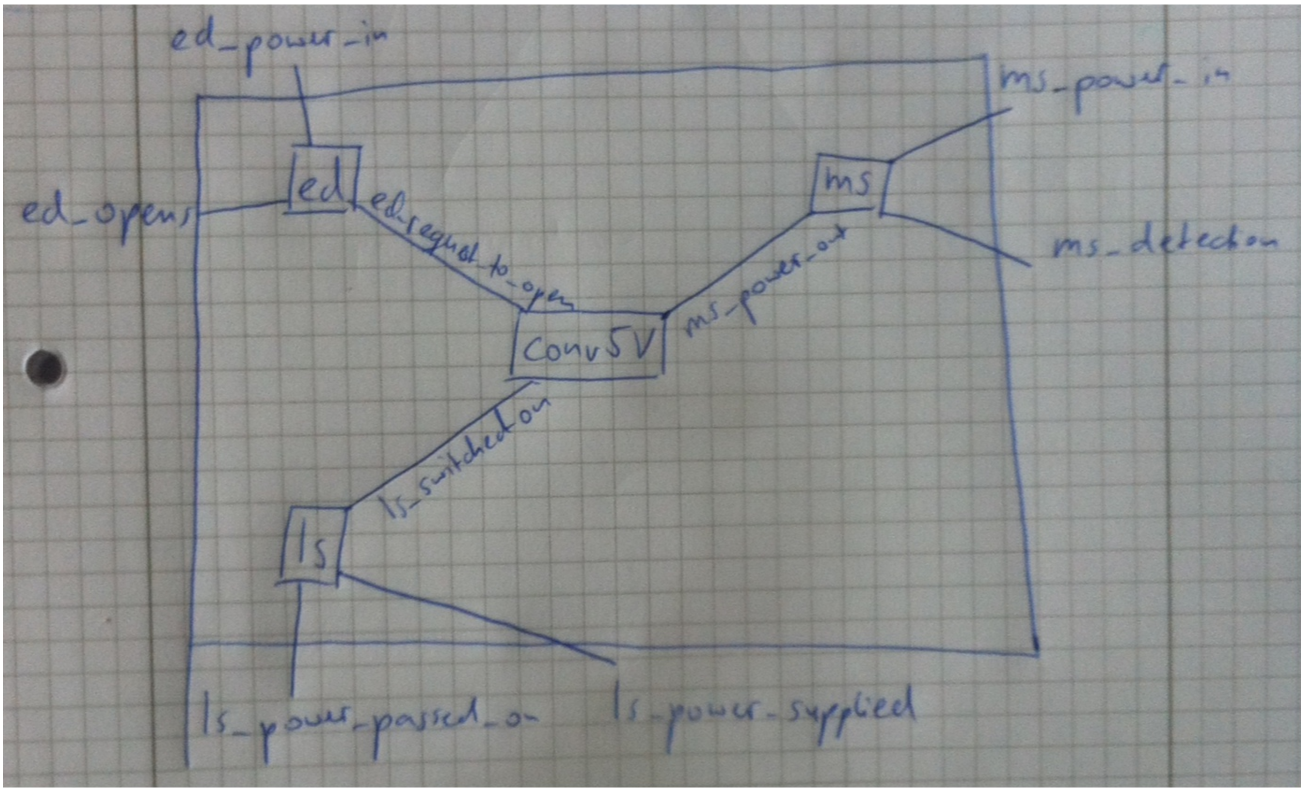
c) I chose the devices motion sensor, electronic door and light switch to create an artifact. The focus in this model is

Modeling goal:

Fragment of reality: ...

Perspective: ...

d) This is the functional network



e) This is my specification of the overall system.

In natural language: If and only if the motion sensor is connected to a 230 volt power supply and it has detected motion in the last two minutes then it sends 5 volt power to the conv5V which enables the electric door to be opened if and only if it is supplied with 230 volt power or the light switch to pass on its power if and only if it is supplied with 230 volt power.

Which gives the formula

ms_power_in
/\
ms_detection

< − >

conv5V

$< - >$

(ed_power_in $< - >$ ed_opens)

\/

(ls_power_supplied $< - >$ ls_power_passed_on)

where Definition conv5V :=

ms_power_out
/\
ed_request_to_open
/\
ls_switched_on

Exercise 4

a) My installation consists of the devices: `motionsensor`, `elektric door` and `light switch`.
These devices are defined as follows:

```
(* Device: Elektronic Door. *)
(* Author: Johan van den Heuvel. *)
Variable ed_power_in : Prop.
  (* 230 Volt  on the input plug of the elektronic door. *)

Variable ed_request_to_open : Prop.
  (* The door receives a signal to open. *)

Variable ed_opens : Prop.
  (* The door opens.*)

Definition elektronicdoor :=

      ed_power_in
    /\
     ed_request_to_open

  <->

    ed_opens
.
  (* The door opens if there is power to the door and the door receives an signal to open *)
```

```
(* Device: motion sensor. *)
(* Author: Thomas Churchman. *)
Variable ms_power_in : Prop.
  (* Voltage in the input plug of the motion sensor. *)
Variable ms_power_out : Prop.
  (* Voltage on the output plug of the motion sensor. *)
Variable ms_detection : Prop.
  (* Movement was detected in the past two minutes. *)

Definition motionsensor :=
      ms_power_in
    /\
      ms_detection
  <->
    ms_power_out
.
  (* There is a voltage on the out port of the motion sensor
     if and only if there is voltage on the in port of the
     motion sensor and movement was detected in the past two minutes.
  *)
```

```
(*Device: Light switch*)
(* Author: Matti Eisenlohr. *)
Variable ls_switched_on : Prop.
(*Light switch is switched on*)
Variable ls_power_passed_on : Prop.
(*The light switch passes the power it receives on*)
Variable ls_power_supplied : Prop.
(*Power is supplied to the light switch*)

Definition light_switch :=
     ls_power_supplied
/\
     ls_switched_on
<->
     ls_power_passed_on
.
(*Power is passed on by the light switch if and only if power is supplied to the switch and th
```

Besides these devices I needed the following connectors:

```
Definition conv5V :=
        ms_power_out
        /\
        ls_switched_on
        /\
        ed_request_to_open.


...
```

And this is the specification of the overall system:

```
Definition sensor_door_light :=
    ms_power_in
    /\
    ms_detection
    <->
        conv5V
        <->
            (ed_power_in <-> ed_opens)
            \/
            (ls_power_supplied <-> ls_power_passed_on)
.
```

And finally, this is my correctness theorem:

```
Theorem myInstallationWorks :
 elektronic_door /\ light_switch /\ motionsensor -> sensor_door_light
.
```